

# Detecting Mass-Mailing Worm Infected Hosts by Mining DNS Traffic Data

Keisuke Ishibashi, Tsuyoshi Toyono, and  
Katsuyasu Toyama  
NTT Information Sharing Platform Labs.,  
NTT Corporation  
3-9-11 Midori-cho, Musashino-shi,  
Tokyo 180-8585, Japan  
{ishibashi.keisuke, toyono.tsuyoshi,  
toyama.katsuyasu}@lab.ntt.co.jp

Masahiro Ishino, Haruhiko Ohshima,  
and Ichiro Mizukoshi  
NTT Communications Corporation  
2-3-5 Otemachi, Chiyoda-ku,  
Tokyo 100-0004, Japan  
{ishinom,ohshima,ichiro}@ocn.ad.jp

## ABSTRACT

The Domain Name System (DNS) is a critical infrastructure in the Internet; thus, monitoring its traffic, and protecting DNS from malicious activities are important for security in cyberspace. However, it is often difficult to determine whether a DNS query is caused by malicious or normal activity, because information available in DNS traffic is limited.

We focus on the activities of mass-mailing worms and propose a method to detect hosts infected by mass-mailing worms by mining DNS traffic data. Our method begins with a small amount of a priori knowledge about a signature query. By assuming that queries sent by most hosts that have sent the signature query of worms have been sent by worm behavior, we detect infected hosts using Bayesian estimation.

We apply our method to DNS traffic data captured at one of the largest commercial Internet Service Providers in Japan, and the experimental result indicates that an 89% reduction of mail exchange queries can be achieved with the method.

## Categories and Subject Descriptors

C.2.3 [Computer Communication Networks]: Network Operations – Network monitoring

## General Terms

Security

## Keywords

Domain Name System, mass-mailing worm, data mining

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*SIGCOMM'05 Workshops*, August 22–26, 2005, Philadelphia, PA, USA.  
Copyright 2005 ACM 1-59593-026-4/05/0008 ...\$5.00.

## 1. INTRODUCTION

The Domain Name System (DNS) is a critical system in the Internet; thus, monitoring its traffic and protecting it from malicious activity are important for security in cyberspace.

In addition, because most applications rely on the DNS when they access the Internet, by monitoring DNS traffic, we can effectively monitor the activity of those applications [20], including malware activity, and protect other Internet resources such as mail servers. For example, MyDoom.A, which attacks the SCO web site on a specific day, can be found by monitoring DNS traffic because it sends queries to the DNS server to resolve the domain name “www.sco.com.” to an IP address, before it attacks the web site [17].

However, for most malware activity, it is often difficult to find such a clear characteristic query which can be used as a signature of the malware activity in DNS traffic. For example, botnets, which are a groups of hosts that can be controlled by a malicious user, recently become a critical security threat to the Internet [4]. It has been reported that by monitoring DNS traffic carefully, botnet activities can be found [6, 7]. However, there is a great variety of botnet activity such as accessing control servers, attacking web sites, or sending spam e-mail, and those activities may not be hard coded in the malware but ordered by a controller through some control channel such as an Internet Relay Chat (IRC). Thus, it cannot be expected to obtain all of the queries by which we can characterize botnet activity. Even if we have characteristic queries of botnets, they may be only a part of the queries sent by botnets, and we cannot expect to detect the bots with a high probability.

As another case, mass-mailing worms, such as Netsky [18], propagate by sending virus e-mails to addresses that are found in the infected host. They send DNS queries to find the mail server of the address, so they leave footprints of their activity in DNS traffic. Specifically, after mass-mailing worms have spread through the Internet, we have observed that the number of queries sent to Internet Service Provider (ISP) DNS servers has increased significantly. However, the same queries cannot be expected to be sent from all infected hosts by this type of worm because there is a variety of target addresses found in the infected host. While reverse-engineering results by anti-virus vendors report some characteristic domain names queried by the worms [16], it may

only be part of the queries sent by those worms, because current worms have the functionality to encrypt their codes.

In those cases, we cannot expect to know all of the characteristic queries of worms with which we can capture their activities or detect infected hosts. Here, if we have some characteristic queries that can be used as signatures of the worm, which may reflect a part of a worm’s activity, and if there are queries that are sent by most hosts that send the signature queries, then those queries can also be assumed to be suspicious worm queries. Thus, based on this naive assumption, there is a possibility of detecting suspicious queries other than the initial characteristic queries and detecting infected hosts that are not found by signature-based detection.

We focus on the activity of mass-mailing worms in DNS traffic, and propose a method to detect hosts infected by those worms with partial prior information about the characteristic queries. For this study, it is critical to calculate the probability that a query is caused by malicious activity. For example, if we simply calculate the probability that a domain name is queried by a worm with respect to the number of hosts who have sent the queries of an available signature, then popular domain names such as “yahoo.com.” may get high probabilities because even infected hosts may send queries for popular domains. To avoid this, we use a Bayesian inference method to calculate the suspiciousness of queries for domain names. We apply our method to DNS traffic data captured at one of the largest commercial Internet Service Providers (ISP) in Japan, Open Computer Networks (OCN), and the experimental result indicates that 89% of mail exchange queries can be reduced with the method.

The rest of the paper is organized as follows. In section 2, we mention related work. In section 3, we give a brief overview of the DNS and mass-mailing worms. Then, in section 4, we propose a method to determine characteristic queries among infected users. We give an evaluation example in section 5.

## 2. RELATED WORK

There is a lot of work that has taken the approach of data mining to detect anomalous hosts or worms [2, 3, 9, 15]. For example, Schultz *et al.* proposed a method to detect new, previously unseen malicious executables (*e.g.* attached files in virus e-mail) using Bayesian inference [15]. As in the usual Bayesian estimation, this method uses training data to make the worm signature. However, for DNS traffic, we do not expect to obtain training traffic data that was actually sent from worm-infected hosts because the objective of the method is to detect infected hosts. Thus, we use available characteristic queries, then select hosts sending those queries, and we consider the queries sent from those hosts to be training data.

Recently, monitoring DNS traffic has attracted much attention, because of the importance of protecting DNS servers. For example, it is reported that most query traffic to a root DNS server is bogus [19], which is mainly due to operational errors of other (recursive) DNS servers. In studies on queries to a local DNS server caused by worm activity, Wong *et al.* analyzed the traffic behavior of mass-mailing worms, including DNS query patterns [22]. They show that there are positive correlations between the number of SMTP flows and volume in DNS traffic, and DNS traffic can be used as signal of mass-mailing worm behavior.

Studies on DNS-based worm detection have also been proposed [13, 21]. In [21], it is assumed that normal Internet access is associated with DNS queries; thus, access without any DNS queries is detected as suspicious access. This method is effective at scanning worms that do not use DNS, but is not applicable to detecting mass-mailing worms. In a similar manner to our work, Musashi *et al.* reported on detecting mass-mailing worms using DNS traffic data in campus networks [13]. They assumed that the hosts that send many mail exchange queries are infected by a mass-mailing worm because normal end hosts use a local mail server to send their mail and do not send such queries. However, in commercial ISPs, there may be hosts running a mail server, so this assumption is not applicable and a more sophisticated investigation is required.

In summary, to the best of our knowledge, this paper is the first one that applies the data-mining approach to DNS traffic and detects hosts infected by mass-mailing worms.

## 3. BACKGROUND

In this section, we describe a brief background of our work, especially the DNS, mass-mailing worms, the DNS traffic data used in this paper, and statistics of the data.

### 3.1 DNS

The DNS provides a service that maps domain names to IP addresses, mail exchangers, and name servers [11]. DNS servers have the mapping data as *Resource Records* (RR). An RR consists of a tuple  $\langle name, TTL, class, type, data \rangle$ , where *name* is the domain name such as *www.acm.org*, *TTL* is time-to-live for the record, which specifies the time within which the cache expires, *class* is Internet (IN) for most records, and *data* is specified data of the record. Defined RR types are listed in Domain Name System Parameters [5]. For example, the IP address of a domain name is called an “A” resource record and a mail exchange server for a domain name is called an “MX” resource record [5]. When a client tries to obtain the record, (*e.g.* to resolve a domain name to an IP address), it sends a query to a local DNS server that specifies  $\langle name, class, type \rangle$ . Hereafter, we define a  $\langle name, type \rangle$  couple as query content (*e.g.* “MX www.yahoo.com”) or simply content. (The *Class* field is omitted because most queries are of the “IN” class)

### 3.2 Mass-Mailing Worms and DNS

Mass-mailing worms propagate by sending virus e-mails to addresses that are found in an infected host. Most of them send e-mail with their own SMTP engine [22]. Before they send the mail, they send MX queries to local DNS servers to find the appropriate mail server for the mail address.

While the mail sent by those worms has a clear signature in the attached files, those worms have their own SMTP engines, and their mail does not rely on the ISP mail servers of infected hosts. Thus, we cannot detect those hosts by monitoring mail server traffic. However, most of them use a local DNS server, whose traffic we can relatively easily monitor.

### 3.3 Data Set

For the experiment, we use DNS traffic data that were captured at one of the largest commercial ISPs in Japan, “OCN”. We captured some of the DNS query packets sent to the DNS cache servers of the ISP. Data were captured from

15:00 to 17:00 on March 7, 2005. There were 31,278,205 queries, the number of unique hosts seen in the period was 221,782, and the number of unique query contents was 1,302,204.

The percentage of each query type is listed in Table. 1. As a comparison, we also list the ratios captured in February 2004 at the same place before the most popular mass-mailing worm, Netsky, appeared [18]. For data captured in March 2005, we found many MX queries. The percentage of MX queries increased from only 2% to about 36%; thus, most of the MX queries are suspected of having been sent from the infected hosts. Since Netsky appeared, we have always see many MX queries with a similar ratio, and the total rate of queries sent to the servers has also been increased.

By observing DNS traffic, we found some characteristic query content sent by hosts who send many MX queries. The most typical query is the one whose query type is not defined. By monitoring the byte string of the queries, we found that this is a format error due to a bug of worms that want to send MX queries<sup>1</sup>. Therefore, hosts who send this query content are clearly judged to be infected by mass-mailing worms. Thus, we used this type of query as a signature query. The number of unique hosts that sent this type of query is 855, (4% of the total number of unique hosts).

**Table 1: Percentage of query types (%)**

Query type	Ratio (Feb. 2004)	Ratio (Mar. 2005)
A	77.15	46.11
MX	2.10	36.34
PTR	15.16	6.18
NS	0.21	0.05
SOA	0.93	0.56
CNAME	0.03	0.00
AAAA	1.72	1.69
ANY	0.86	0.14
SRV	1.10	0.22
Other	0.75	8.70

## 4. PROPOSED METHOD

The objective of the method is to detect a worm-infected host by using queries sent by the host. In other words, for each host  $h$ , given the query content sent by a host  $h$  as  $Q_h$ , we want to calculate

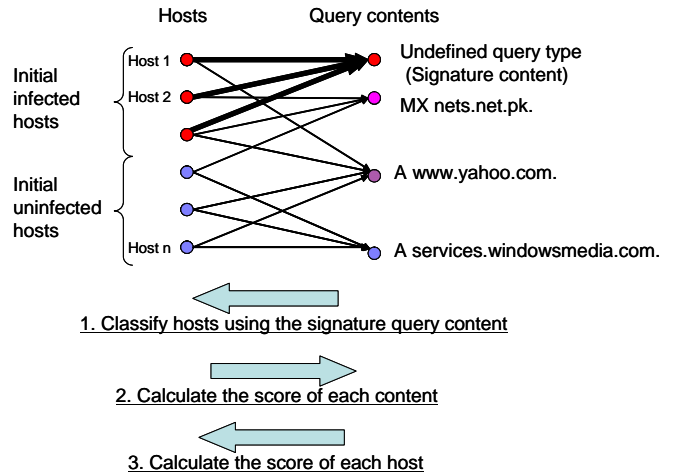
$$\Pr(\text{host } h \text{ is infected} | Q_h). \quad (1)$$

To calculate the probability, we propose a three-step procedure starting with the signature query content found in the DNS traffic data, as shown in Fig. 1 (the arrows in the three steps mean that the procedures are done with the directions of each arrow<sup>2</sup>)

In the following equations, we borrow the ideas used in Bayesian spam filtering [1, 8, 14].

### 4.1 Classifying Hosts

<sup>1</sup>In the DNS question section, the end of the query name part is indicated by a null byte, and the query type part follows [12]. However, in those queries, the null byte is accidentally inserted into the query name section. Thus, the bytes after the null byte are parsed as the query type, which is not defined.



**Figure 1: Host and query content pair**

The first step is to classify each host based on whether the host sends the signature query content or not.

Let  $H = \{h\}$  be the set of total hosts that have sent more than one query in a traffic-monitoring period, and let the set of those hosts that have sent the signature content be  $I$ . That is,

$$I := \{h \in H | h \text{ sent signature query content}\},$$

and call host  $h$  in  $I$  the initially infected hosts.

### 4.2 Scoring Query Content

The second step is to calculate the score of a query content that expresses the infection probability of a host given that the host sends the query content.

There are two ways of performing the calculation: host-based and query-based.

#### 4.2.1 Host-Based Scoring

Host-based scoring calculates the score of a query content based on the number of initial infected hosts who send the content. In this method, we first calculate ratio for each query content  $q$  as

$$I_H(q) = \frac{\#\{h \in I | q \in Q_h\}}{\#I}. \quad (2)$$

This value indicates the ratio of content  $q$  that is queried by initially infected hosts. As described above, because even infected hosts may send queries to popular domain names, the popular query content can also obtain a high  $I_H(q)$ . Thus  $I_H(q)$  is not appropriate to estimate the suspiciousness of the query content  $q$ . To avoid this, we also calculate the likelihood that content is queried by initially uninfected hosts.

$$N_H(q) = \frac{\#\{h \in \bar{I} | q \in Q_h\}}{\#\bar{I}} \quad (3)$$

Then, the score of query  $S_H(q)$  is calculated as follows.

$$S_H(q) = \frac{I_H(q)}{I_H(q) + N_H(q)} \quad (4)$$

$S_H(q)$  is a rough estimate of the probability that a host sending a query “ $q$ ” is infected by a mass-mailing worm. Actually, to calculate the probability, we need to know the probability that a host is infected, but because we do not have a priori knowledge of the probability, we set the probability to 0.5 and obtain the above equation.

With this calculation, a query sent by only one host is scored as 1 or 0, depending on whether the host is an initially infected host or not. However, we cannot expect that such queries truly have such a high or low probability. Thus, as in [14], we modify the probability using constants  $P_{init}$  and  $N_{init}$  as follows.

$$S'_H(q) := \frac{P_{init} + N_q * S_H(q)}{N_{init} + N_q} \quad (5)$$

where  $N_q$  is the number of total queries for the query content.

As can be seen, the initial score of a content is  $P_{init}/N_{init}$ , which indicates the a priori probability that a host is infected given that the host sends the query content. As the number of queries for the content increases, the score  $S_H(q)$  is weighted in the calculation of  $S'_H(q)$ . Because we do not have any a priori knowledge other than the signature content, we set  $P_{init}$  to 0.5, and  $N_{init}$  to 1.

#### 4.2.2 Query-Based Scoring

For host-based scoring, the number of queries sent by the same host is ignored. We only consider whether or not there is a query for the name of a host. In query-based scoring, we take the number of queries sent by the same host into account. In other words, while host-based scoring uses the number of links in Fig. 1 to calculate the score, query-based scoring uses the number weighted by the number of queries for the link. The score is calculated as follows.

$$I_Q(q) = \frac{\sum_{h \in I} \text{number of queries of “} q \text{” sent by host } h}{\sum_{h \in I} \#Q_h} \quad (6)$$

$$N_Q(n) = \frac{\sum_{h \in \bar{I}} \text{number of queries of “} q \text{” sent by host } h}{\sum_{h \in \bar{I}} \#Q_h} \quad (7)$$

Similarly, the score of  $q$ ,  $S_Q(q)$  is calculated as follows:

$$S_Q(q) = \frac{I_Q(q)}{N_Q(q) + I_Q(q)}. \quad (8)$$

For query-based scoring, we apply the score adjustment as in equation (5).

### 4.3 Scoring Hosts

The method then calculates the score of a host that indicates the probability that the host is infected based on both the queries sent by the host and the scores of the queries.

There are various heuristic ways to calculate the score of the host. It is reported that using only queries with extreme scores (near 1 or 0) and taking the geometric mean of the scores performs well for Bayesian spam filtering [1, 8]. We thus set two thresholds,  $T_H$  and  $T_L$ , for high values and low values, respectively, and calculated the host score as follows. Here we only show the calculation using host-based scoring, but using query-based scoring can also be calculated. Let

$m = \#\{q \in Q_h | S'_H(q) > T_H\}$  and

$$I(h) = \begin{cases} 1 & \text{if } m = 0 \\ 1 - (\prod_{\{q \in Q_h | S'_H(q) > T_H\}} S'_H(q))^{1/m} & \text{otherwise} \end{cases} \quad (9)$$

Let  $k = \#\{q \in Q_h | S'_H(q) < T_L\}$  and

$$N(h) = \begin{cases} 1 & \text{if } k = 0 \\ 1 - (\prod_{\{q \in Q_h | S'_H(q) < T_L\}} (1 - S'_H(q)))^{1/k} & \text{otherwise} \end{cases} \quad (10)$$

Finally, the degree of belief that host  $h$  is infected is calculated as follows [8]:

$$P(h) = \frac{1 + N(h) - I(h)}{2(N(h) + I(h))} \quad (11)$$

Like the query score, the above score also takes the value between 0 and 1, and 1 means that the host is infected with a probability of one.

Finally, we judge a host  $h$  is infected if the  $P(h)$  is larger than a predefined threshold.

## 5. EXPERIMENTAL RESULTS

In this section, we describe the experimental results of our method on actual DNS traffic explained in section 3.3.

First, in Table 2, we list the top 20 contents according to their host-based scores. By definition, the signature query gets the highest scores, while other queries have similarly high scores. Almost all of these are MX queries.

Query content with ranks two, three, or five are domains of mail addresses for customer support service of a company [10] and were found to be used as the sender address of some worms. In addition, we also have MX queries for major ISPs in this ranking. Thus, sending MX queries is not normal behavior because hosts that send queries to the DNS server are mainly consumer hosts. This ranking indicates that those hosts are suspicious. However, there is some content such as “MX m.” that is clearly expected to be queries sent by worms. These are considered as mistakes of the worms in finding or creating target mail addresses. By mining traffic data from a signature query (undefined query type), we selected these suspicious query contents from the 1,302,204 query contents.

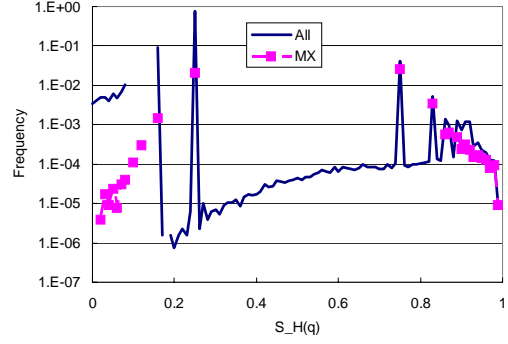
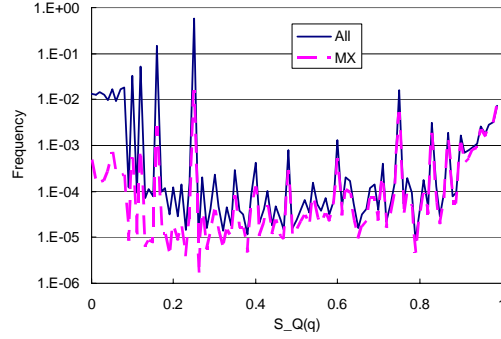
In addition to the query contents in the table, The query content of “A www.yahoo.com” has a score of 0.53. This means that while the content is queried by many infected hosts, uninfected hosts send queries as well. Therefore, the score of the query becomes neutral, that is, the content gives no information about whether a host is infected or not. In addition, the result indicates that even if a worm sends a lot of normal query traffic to confuse the algorithm, it may be filtered because that query content will be given normal scores.

As a comparison, we show the queries ranked by the number of infected hosts that sent the query (Table 3). As can be seen, domains with a popular web site are highly ranked among the suspicious content listed in table 2. Thus, we cannot detect suspicious queries by simply monitoring the queries sent by hosts that send the signature query.

The frequency plots of host-based scores are shown in Figure 2. There are several spikes, especially on scores 0.75 and 0.25. It is clear from Eq.(5), there is only one initial infected (uninfected) host that sent the query, then the content is scored as 0.75 (0.25).

**Table 2: Host-based score rank ( $S'_H(q)$ )**

Rank	$S'_H(q)$	Query content
1	0.999	Undefined query type
2	0.997	MX nets.net.pk.
3	0.997	MX gto.net.om.
4	0.995	MX sexnet.com.
5	0.994	MX lebanon-online.com.lb.
6	0.993	MX aa2.so-net.ne.jp.
7	0.993	MX domain.com.
8	0.992	MX m.
9	0.992	MX phx.gbl.
10	0.991	A dev.null.
11	0.990	MX -.
12	0.990	MX ocn.ad.jp.
13	0.990	MX hatch.co.jp.
14	0.990	MX ezweb.ne.jp.
15	0.990	MX a.
16	0.990	MX rcpt-impgw.biglobe.ne.jp.
17	0.990	MX nifty.ne.jp.
18	0.990	MX 2.
19	0.990	MX nifty.com.
20	0.990	MX h.

**Figure 2: Frequency of query score (host-based)****Figure 3: Frequency plot of query score (query-based)****Table 3: Queries ranked by  $I_H(q)$** 

Rank	$I_H(q)(\%)$	Query content
1	100.00	Undefined query type
2	31.92	MX nets.net.pk.
3	31.43	MX gto.net.om.
4	28.50	A img.yahoo.co.jp.
5	28.34	A ai.yimg.jp.
6	27.85	MX sexnet.com.
7	26.71	A pa.yahoo.co.jp.
8	24.76	A www.yahoo.co.jp.
9	20.20	A i.yimg.jp.
10	19.71	MX lebanon-online.com.lb.
11	15.96	A ca.c.yimg.jp.
12	15.47	A search.yahoo.co.jp.
13	15.47	A rd.yahoo.co.jp.
14	14.82	A srd.yahoo.co.jp.
15	14.33	A dailynews.yahoo.co.jp.
16	13.52	MX domain.com.
17	12.54	A shopping.yahoo.co.jp.
18	11.73	A headlines.yahoo.co.jp.
19	11.07	A wpad.
20	10.59	MX ezweb.ne.jp.

We also plot the frequency of the MX query type. We can see that MX queries tend to have high scores. That is expected from Table 2.

We also calculate the query-based scores (Fig. 3). The shape of spikes in the graph are roughly similar to those in Fig. 2. Thus, hereafter, we only use the host-based score, but combining the two scoring methods can be considered to be effective if the number of queries are a good sign of worm propagation.

We then calculated host scores using Eq. (11). We choose host-based scores in this paper and set  $T_U$  to 0.95 and  $T_L$  to 0.05. A frequency plot of host scores is shown in Fig. 4. It is clearly seen that there is a high spike at 0.5. In Eqs. (6)–(11), hosts that do not send a query whose score is larger than  $T_U$  or smaller than  $T_L$  obtain the value. In that case, we judge the host to be a neutral host, so a score of 0.5 is reasonable. If we change the values of  $T_U$  and  $T_L$ , there is a trade-off between false positives and false negatives. That is, false positives increase when  $T_U$  decreases while false negatives decrease. Future work is to find appropriate values of those parameters.

We also plot the frequency of host scores that have sent signature queries. The difference between the two lines in the high-score area indicates the number of hosts that can be detected by our method, but not detected by the signature method. For example, those hosts that sent the queries listed in table 2 except the signature query can be detected

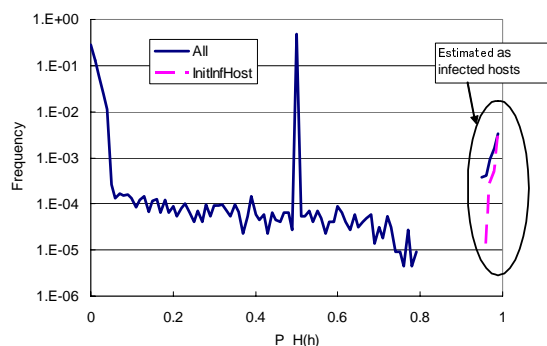


Figure 4: Frequency plot of host score

only by our method.

If we set the threshold score to 0.95, then the number of MX queries sent by detected hosts is 10,095,362, which is actually 89% of the total number of MX queries. Thus, by filtering MX queries from those hosts, there is a possibility to reduce useless MX queries by 89%. Because we do not have a correct infected hosts list, it is possible that there are uninfected hosts with scores larger than 0.95, and simply applying our method may filter the queries from those hosts. However, in actual operation in the ISP, automatic filtering tends to be avoided, but filtering user packets is mainly done manually. In that case, our method helps operators in detecting infected hosts.

## 6. CONCLUSION AND FUTURE WORK

We proposed a method to detect hosts infected by mass-mailing worms using DNS traffic data. We used a Bayesian inference method to detect hosts with partial knowledge of query traffic of those worms. While we use a bug of a worm in this paper, the method is not limited to bugs or format error queries; it can be applied to queries for domain names that are found by reverse-engineering of the worms by anti-virus vendors.

Experimental results indicate that a reduction of 89% of MX queries may be achieved by our method. As the work is in progress and the results described in this paper are preliminary, we will evaluate our method according to its sensitivities of parameters and the dependencies of the results on the used data set. In addition, evaluation of misclassification, e.g., false alarm rate, remains for future work.

## 7. REFERENCES

- [1] P. Graham, "A Plan for Spam," <http://www.paulgraham.com/spam.html>.
- [2] A. Gupta and R. Sekar, "An Approach for Detecting Self-Propagating Email Using Anomaly Detection," Proc. Recent Advances in Intrusion Detection, September 2003.
- [3] H. Han, X. L. Lu, J. Lu, C. Bo, and R. Yong, "Data Mining Aided Signature Discovery in Network-based Intrusion Detection System," Proc. ACM SIGOPS'02, 2002.
- [4] The HoneyNet Project & Research Alliance, "Know Your Enemy: Tracking Botnets," <http://www.honeynet.org/papers/bots/>.
- [5] Internet Assigned Number Authority, Domain Name System Parameters, <http://www.iana.org/assignments/dns-parameters/index.html>.
- [6] J. Kristoff, "Botnets," NANOG 32, October, 2004.
- [7] J. Lick, "Tracking A Zombie Army," APRICOT 2005, February, 2005.
- [8] G. Louis, "Bogofilter Calculations: Comparing Geometric Mean with Fisher's Method for Combining Probabilities," <http://www.bgl.nu/bogofilter/fisher.html>.
- [9] M. V. Mahoney and P. K. Chan, "Learning Nonstationary Models of Normal Network Traffic for Detecting Novel Attacks," Proc. ACM SIGKDD'02, 2002.
- [10] Microsoft Office XP Developer Technical Support, <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/modcore/html/demscmodpss.asp>.
- [11] P. Mockapetris, "Domain Names - Concepts and Facilities," RFC1034, November 1987.
- [12] P. Mockapetris, "Domain Names - Implementations and Specifications," RFC1035, November 1987.
- [13] Y. Musashi, R. Matsuba, and K. Sugitani, "Indirect Detection of Mass Mailing Worm-Infected PC terminals for Learners," Proc. ICETA2004, 2004.
- [14] G. Robinson, "A Statistical Approach to the Spam Problem," Linux Journal, no. 107, March 2003.
- [15] M. G. Schultz, E. Zadok, and S. J. Stolfo, "Data Mining Methods for Detection of New Malicious Executables," Proc. IEEE Symposium on Security and Privacy, May 2001.
- [16] Symantec Corp. W32.Beagle.a@mm, <http://securityresponse.symantec.com/avcenter/venc/data/w32.beagle.a@mm.html>.
- [17] Symantec Corp. W32.Mydoom.A@mm, <http://securityresponse.symantec.com/avcenter/venc/data/w32.novarg.a@mm.html>.
- [18] Symantec Corp. W32.Netsky.P@mm, <http://securityresponse.symantec.com/avcenter/venc/data/w32.netsky.p@mm.html>.
- [19] D. Wessels and M. Fomenkov, "Wow, That's a Lot of Packets," in Proc. 2003 Passive and Active Measurements Workshop, April 2003.
- [20] C. E. Wills, M. Mikhailov, and H. Shang, "Inferring Relative Popularity of Internet Applications by Actively Querying DNS Caches," Proc. ACM Internet Measurement Conference 2003, October 2003.
- [21] D. Whyte, E. Kranakis, and P. C. van Oorschot, "DNS-based Detection of Scanning Worms in an Enterprise Network," Proc. NDSS'05, 2005.
- [22] C. W. Wong, S. Bielski, J. M. McCune, and C. Wang, "A Study of Mass-mailing Worms," Proc. ACM CCS 2nd Workshop on Rapid Malcode (WORM04), October 2004.